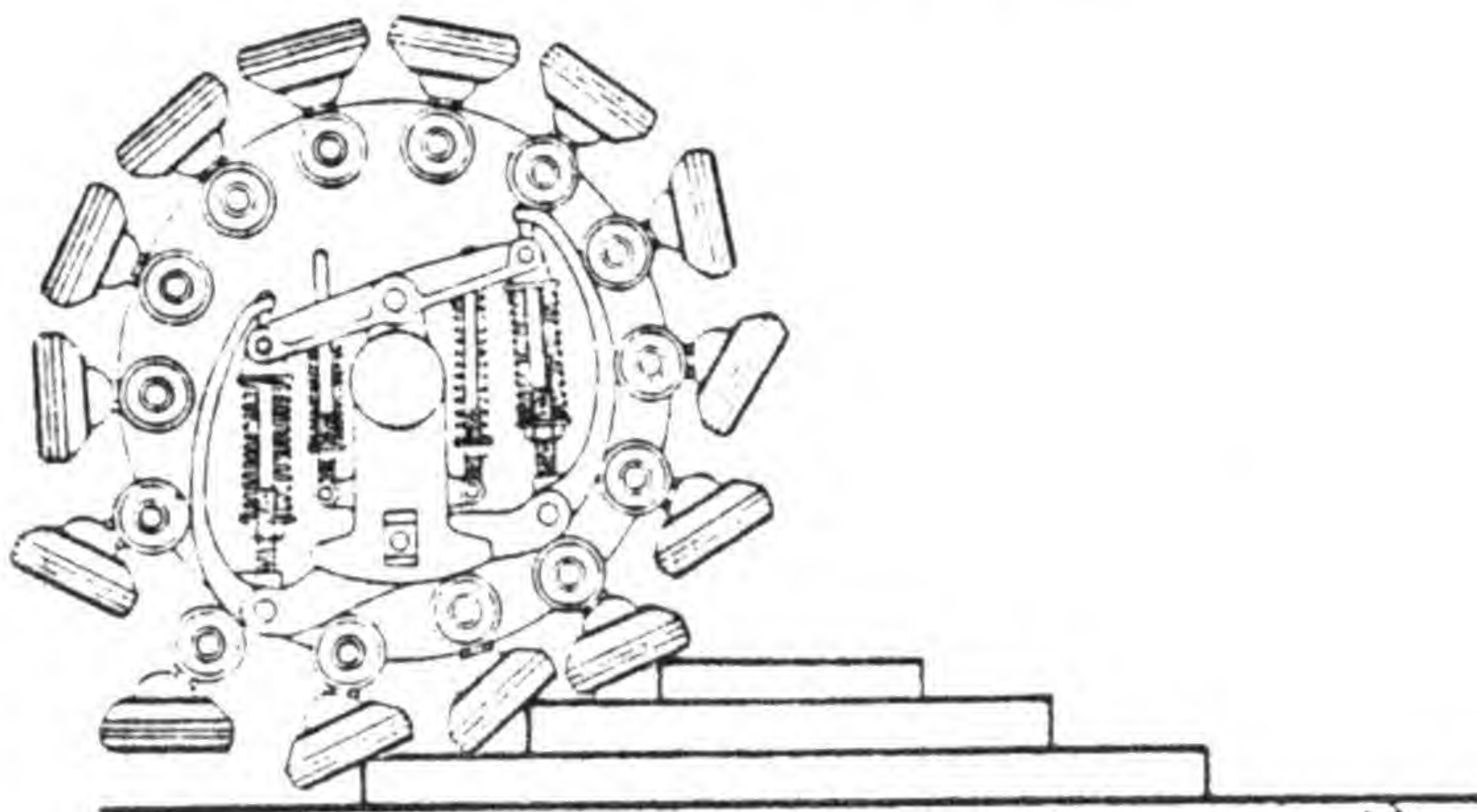


SOFTWARE CPUtm

SUPER STEP

Z80 Processor Model



BY
ALLEN GELDER

© 1980

16K LEVEL II



EMU 02: Software emulation of the 6502 microprocessor. TBUG displays byte, EMU takes it from there. Now you can write, debug and execute 6502 programs on your TRS-80.

- Disassembler posts 6502 mnemonic in scrolling field.
- Single-stepper displays 6502 Processor Model, stack, flag status in before/after form.
- 4-Speed TRACE mode animates 6502 models, activates a keyboard scan port accessible to 6502 instructions.
- Fast interpretive RUN mode for realistic execution.
- Implicit keypad with Backspace, Relative space, more.

How to have a 6502 without having a 6502! Compare, contrast, learn a powerful programming language distinct from Z80 or BASIC, read Apple, PET code. A 6502 Software CPUtm 16K Level II TRS-80, TBUG required. No. BL-1 \$24.95



Super TLEGS: Onboard relocater for T-BUG, moves T-BUG to any desired RAM location. Now examine, modify, # P punch backup copies of formerly coincident program material, experiment with parallel kustom T-BUGs. Also will relocate TSTEP.

16K Level II Super TLEGS No. LL-0

9.95

Include .75 each
postage, CA add 6%



ALLEN GELDER SOFTWARE
Box 11721 Main Post Office
San Francisco, CA 94101

TRS-80, TBUG tm Radio Shack/Tandy Corp.
Software CPU tm Allen Gelder Software.

ABSTRACT: Super STEP is a machine language program designed to be used in conjunction with TBUGtm (No. 26-2001), the TRS-80tm Z80 monitor program sold by Radio Shack/Tandy Corp. Super STEP displays, in various formats, a scrolling field of disassembled RAM locations corresponding to the PC of an animated Z80 Processor Model. The Z80 Model can single-step or TRACE any Z80 machine code, and is displayed in a before/after format that includes CPU registers, stack elements and an expanded flag register configuration. Also displayed is an intelligent RAM window that selectively posts local RAM environments or a user-designated area. One or both of the Z80 Models can be suppressed, as can the scrolling field, for full screen access. The variable speed TRACE mode has a user HALT, a foreground breakpoint (76 HALT) and a dynamic SKIP key for user control during the TRACE. An implicit keypad is opened under several control points, with keys that variously service the display or assist local editing in RAM; also faster tape I/O is available. Super TLEGS (No. LL-0) will relocate Super STEP for total address space access.

LOADING	2
DISPLAY, FIG.1.	3
KEYPAD, FIG 2.	5
KEYS BY ROW	6
KEYS BY FUNCTION	7
MISCELLANEOUS	14
DANGEROUS BENDS	15
STELEGS RELOCATOR	16

TBUG, TRS-80tm Radio Shack/Tandy Corp. Super STEP, Software CPUtm Allen Gelder Software. All Rights Reserved.

LOADING: Load Super STEP under the SYSTEM command. First TBUG.

SYSTEM	
*? TBUG	Load TBUG.
*? SPRSTP	Load Super STEP. Volume 5½.
*? /	<u>ENTER</u> Transfer control to 19456.
# S	The S key links TBUG and Super STEP.

Alternatively,

SYSTEM	
*? TBUG	
*? /	<u>ENTER</u> Transfer control to TBUG.
# L	Load SPRSTP using the TBUG # L command.
# J 4C00	Jump to cold start entry pt.
# S	Link with the S key.

Super STEP is now in place from 4B00-68FF. (On versions relocated by Super TLEGS linking proceeds in the same two steps; # J 4C00 or relocated equivalent, then # S for final linking.) The # S key brings up the copyright header and the TBUG # prompt character.

AFTER

BEFORE

0 3F07 5F
01 1302
0000 0000

AF' = 0000 0000 = 0C' 0000 0000

IE' = 0000 0000 = 0L' 0000 0000

AF = 0400 0000 = 0C' 0400 0000

IE = 0000 7005 = 0L' 0000 7005

IX = 0000 0000 = 0Y 0000 0000

SP = 6900 7004 = PC 6900 7003

SZ-H-PC 00000000

00000000

P NZ PO NC P NZ PO NC

6900 FF 7003 02 6900 FF 7003 02

6901 FF 7004 77 6901 FF 7004 77

6902 FF 7005 04 + 6902 FF 7005 00

6903 FF 7006 00 6903 FF 7006 00

6904 FF 7007 00 6904 FF 7007 00

0 H 7000 DE LD A=02H

7001 02

7002 06 ADD A=02H

7003 02

7004 77 LD (HL),A

7005 04

RAM location and byte contents.

Disassembled listing. Disassembler follows program flow order or straight line.

Topmost five stack elements.

RAM environment selected by Intelligent RAM window.

RAM Window. Here in the ← intelligent mode, has selected current HL register to post the register indirect load.

Flag expansion. Bit assignment header.

Bit expansion.

Assembly mnemonic for testable bits.

CPU registers.

PC corresponds to the most recently executed instruction.

FORMATTING THE DISPLAY: The display normally operates in a split screen mode; the left 1BH columns are scrolled and the right 25H columns are not scrolled (except during certain block RAM displays). Both sides may be reformatted as follows:

SCROLLING: As loaded, full left scrolling of memory locations and contents plus the disassembled listing. Alternatively, the user can select a reduced scrolling mode. (See use of the = key, below.) In this mode only two locations are scrolled, at the bottom of the left side of the screen. The scrolled areas are two lines 1BH characters long. The starting screen locations for these lines are located inside Super STEP, in RAM locations 651C and 651E. The user may thus vector these lines as desired to avoid conflict with screen material displayed by a subject program.

WORKSPACE: As loaded, the workspace displays two Z80 Processor Models, corresponding to prior and present CPU status relative to the instruction found at the PC of the labeled Model. The unlabeled model represents CPU status relative to its PC. This arrangement unambiguously shows the effects of each instruction. The unlabeled, before Model may be suppressed. (See → key, below.) Also, both models may be suppressed. (See - key, below.)

RAM WINDOW: As loaded, the RAM Window is in the automatic mode; the user may select any RAM area of interest for observation. (See ← key, below.)

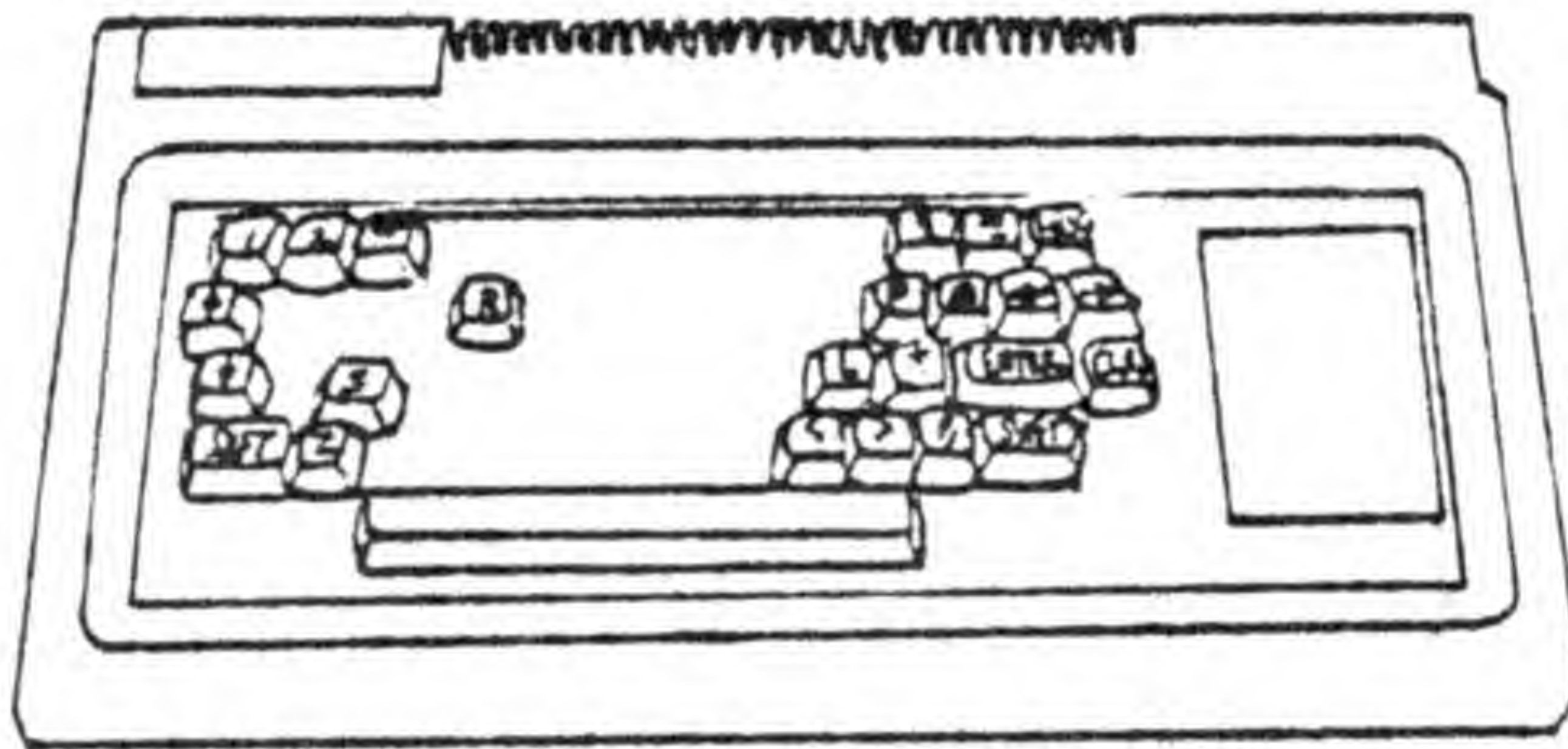


Fig. 2. Implicit keypad under SPRSTP

CONTROL POINTS: There are five control access points.

Control point 1: Open when TBUG # prompt character is displayed.
like # R, # M, etc.

Control point 2: Open under the TBUG # M command, just after
user entry of a two byte RAM address.

M nnnn bb

This is the access point shared by the modify memory functions of TBUG. Most SPRSTP keys are accessible here, including SPACEBAR and : TRACE.

Control point 3: Open under : TRACE mode to accept speed change, / SKIP, and Z-HALT.

Control point 4: Open under (SHIFT) R to accept hex-digit values, ←left and →right cursor, and X exit the mode keys.

Control point 5: Opens under alternate ← RAM Window mode keystrokes to accept a two byte RAM address or X exit the mode. The value entered defines the user RAM Window environment.

KEY FUNCTIONS: By row, from the bottom:

SPACEBAR	CP2	Single-steps current instruction.
Z	CP3	HALT key under : TRACE.
(SHIFT) <	CP2	Delete byte, move string to FFFF one up.
(SHIFT) >	CP2	Insert byte, move string to FFFF one down.
/	CP2	Suspends Z80 Model activity, makes dis- assembler straight-line.
/	CP3	SKIP current instruction under : TRACE.
(SHIFT) ?	CP2	Display ASCII equivalent of current byte.
↓	CP2	Displays relative location and byte contents.
(SHIFT) ↓	CP2	Relative Space memory advance.
S	CP1	Brings up copyright, links TBUG and SPRSTP.
(SHIFT) L	CP1	# L loads faster tapes made by (SHIFT) P.
;	CP2	Display hex/ASCII 16 character line with checksum, scroll workspace.
(SHIFT) +	CP2	Alternates ; key between hex and ASCII.
ENTER	CP2	Advances memory display. (TBUG)
CLEAR	CP1	Clears current scrolling field.
CLEAR	CP2	Clears workspace area.
↑	CP2	Backspace memory advance.
(SHIFT) ↑	CP2	Return to Reference location.
(SHIFT) R	CP2	Change registers. Opens cursor over AF register. User may enter byte value or advance cursor with → or ←. Exit with <u>X</u> .
(SHIFT) P	CP1	# P punches faster tapes.
@	CP2	@@s the Z80 Models.
←	CP2	Changes RAM Window status.
→	CP4	Cursor right under (SHIFT) R register change.
→	CP2	Alternately suppresses/returns unlabeled Model.
←	CP4	Cursor left under (SHIFT) R register change.
1	CP3	Slow speed under : TRACE.
2	CP3	High speed under : TRACE.
(SHIFT) #	CP1	Loads SPRSTP Models with TBUG register contents.
	CP2	Same under Control point 2.
:	CP2	TRACE until Z-HALT or encountering 76 HALT.
(SHIFT) *	CP2	CALL/RST status. Alternately single-step or directly execute CALLs and RSTs.
-	CP2	Alternately suppresses Workspace display.
(SHIFT) =	CP1	Change scrolling mode from full to reduced, back.
BREAK	CP1	Delink TBUG and SPRSTP.

IMPLICIT KEYPAD: By control point and function.

CONTROL POINT 1: Open when TBUG prompt character comes up.

- # S Links TBUG and SPRSTP and brings up the copyright notice, which will scroll up and out. Initializes the scrolling field and SPACEBAR.
- # BREAK Delinks TBUG and SPRSTP, used before relocating TBUG with Super TLEGS (No. LL-0) relocater.
- # CLEAR Clears the scrolling field and reinitializes the prompt character position.
- # (SHIFT) = Reformats the scrolling field. As loaded all rows scroll in a 1BH wide column. Pressing # (SFT) = reduces the number of scrolling rows to two, located at the bottom left of the screen. The initial positions of the 1BH long scrolled lines are found in RAM locations 651CH and 651EH. The user may change these bytes to a desired screen position from 3C00H to 3FE4H, or suppress one by vectoring into unused RAM. This flexible reformatting is necessary because SPRSTP will sometimes be used to Single-step or : TRACE subject programs that conflict with background display.
- # (SHIFT) # Use of this key results in the loading of the Z80 Models displayed by SPRSTP with the contents of the # R registers displayed by TBUG, and is related to a useful procedure as follows: Suppose you are examining some subject program and wish to initialize SPRSTP relative to some point internal to the subject program. You might start at the entry point of the subject material and Single-step or : TRACE to the desired point. Another way involves using (SHIFT) #. Place a TBUG breakpoint at the desired place in the subject program using # B nnnn. The # J jump to the entrypoint and emerge via the breakpoint. At this point key (SHIFT) # and # F, then # M nnnn, where nnnn is the (now restored) breakpoint location. SPRSTP is initialized relative to the subject program except for the stack; the SP is not transferred so the user may maintain an independent stack. Updating the SP involves finding the value with the # R command and writing it into SPRSTP's SP via the (SHIFT) R register change procedure.

Control Point 1 continued:

- # (SHIFT) P This key is a faster # P punch of cassette tapes obtained by adjusting timing loops. Tapes made under this command will not load under SYSTEM; they are compatible only with the SPRSTP command.
- # (SHIFT) L This key is the # L load command for tapes made by # (SHIFT) P. Note that the normal # P and # L commands are still available.

You should probably make a backup copy of SPRSTP, preferably at the higher baud rate. To do this you will need an independent SYSTEM tape maker. The most versatile is a version of TBUG relocated by Super TLEGS (NO. LL-0). Super TLEGS will also relocate SPRSTP. Here we assume you have a version of TBUG at 7380-7980H. Control is with a 4380-4980 TBUG linked with SPRSTP.

# (SHIFT) P 4B00 68FF 4C00	# Punch with fast loops, then BREAK.
# J 73A0	# J jump to the high TBUG.
# P 4380 4980 43A0	# Punch a SYSTEM loading copy of the low TBUG. This TBUG will still have the fast timing loops from recording SPRSTP.

(It's a good idea to make sure the low copy of TBUG is located first on the tape.) To load the melange, load the low copy of TBUG (which happens to have the fast timing loops) under SYSTEM, then enter TBUG and key (SHIFT) L to load SPRSTP. It's an odd fact that fast loading tapes can be more reliable in that the chance of tape dropout is lessened when less tape is used.

Control point 2: This access point is opened under the TBUG # M command, after user entry of a two byte RAM address. TBUG then displays the contents byte at that address and accepts any of ENTER, a user entered byte value or X, exiting the # M mode. This is the most powerful control point that a monitor can have, because it is at this control point that RAM locations are examined and modified. Both the Single-step and the : TRACE modes are available here, giving the user the ability to examine, modify and execute instructions from the same control point. Why? So you can write, debug and execute machine code with the fewest number of keystrokes. This feature can be very appealing after using multiple pass assemblers and other programming aids that operate with heavy keystroke overhead. Control point 2 is the closest you can get to your RAM and CPU if distance is defined as "number of keystrokes to get something done".

SPACEBAR Use of this key under the TBUG # M command will advance the TBUG displayed memory location and contents like ENTER, but with the desirable addition of a disassembled mnemonic form of the instruction appearing in the scrolled field, and, after the trailing byte of the instruction has been SPACEd, the corresponding action in the Z80 Processor Model. This greatly decreases the imaginative overhead of the programmer, particularly in the extended, before/after Z80 Model format, where side-by-side comparison assures us of our results. Note well that SPRSTP is a live program; all of these instructions are actually being executed. Since there are no write-protect constraints, use of this key can contaminate working RAM or result in fatal instruction executions. On the other hand, all you risk is having to reload.

: TRACE This key initiates automatic single-stepping, starting at the displayed instruction and proceeding, in program

flow order, down to the user-placed 76-HALT string delimiter. 76
 HALT is the foreground breakpoint, and use of the TBUG CD 8043
 # B placed breakpoint will only lead : TRACE flow into TBUG.
 (Although SPRSTP will run TBUG perfectly well; try it with an
 independent, Super TLEGS relocated copy. It helps to use the
 restricted scrolling mode. Line up over the entry point:

M 73A0 ED : TRACE

Exit the : TRACE by using Z-HALT. Note that the conflict
 between the SPRSTP and the subject TBUG scrolling fields can
 be resolved by formatting SPRSTP to say, 3F90 and 3FD0H. Recall
 this is done by changing location 651C and 651EH.) A further
 control access point is opened under : TRACE, with these keys:

- 1 Slow speed : TRACE, posts "1ST" on status panel.
- 2 Fast speed : TRACE, posts "2ND".
- / SKIP key causes next instruction to be disregarded
 by execution flow, typically used to fall out of
 loops during : TRACE. SKIPPed instruction is
 disassembled normally.
- Z Z-HALT provides keyboard HALT control during : TRACE.

Maximum : TRACE speed is obtained under the following conditions;
 Reduced scrolling format and suppressed Z80 Models while (SHIFT) *
 CALL/RST status is set for direct execution.

/ This key stops execution of instructions during the
 Single-step or : TRACE modes. A "/" is posted on the
 status panel. The disassembler becomes straight-line.
 SPACEBAR and : TRACE now operate just the scrolled field,
 useful when only disassembly is needed. To reactivate
 the models press / again.

(SHIFT) R Change SPRSTP registers, turns on a cursor over the
 A register. Entering a byte value will advance the
 cursor. The cursor may be advanced left or right
 by use of the ← or → keys, respectively. This action
 is wraparound; PC is next to AF'. Exit the mode
 by pressing the X key to return to control point 2.

← This key alters the status of the RAM Window area. As loaded, the RAM Window automatically changes under certain instructions to show RAM interaction with the CPU. All instructions that refer to RAM in any way will activate the window during Single-stepping with SPACEBAR or during : TRACE. The alternate display status of the RAM Window is as a window into a user-designated area. Pressing ← will open a control point over the top of the window structure; the user is expected to enter a two byte memory location or to exit via X to control point 2. The location posted in this way will persist through Single-stepping or TRACE to show a continuously updated picture of the designated RAM environment. As loaded this location is set to 5126H, the "execution block" area internal to SPRSTP. The next time you press ← the RAM Window will be returned to the automatic mode. During this intelligent mode the user's screen will contain a simultaneous before/after picture of the RAM as referenced by the PC, the status of the CPU architecture and the RAM interacting with CPU. This is the TRS-80 of the programmer's imagination.

Formatting the Workspace: Under control point 2, we have...

→ Suppress the unlabeled Z80 Model. The next time you press → the model will return. The model is re-initialized when it returns; it does not retain any usable "before" state information.

- Suppress both Z80 Models. In this case both before and after state information can be recovered because the Processor Models are still active (subject to → and /), they just aren't shown.

CLEAR Clears the workspace area. CLEAR under control point 1 will clear the scrolling field.

- (SHIFT) * This key controls whether CALLs and RSTs are to be single-stepped or directly executed. As loaded these instructions will be single-stepped, when (SHIFT) * is pressed a "*" will light in the status panel and CALLs and RSTs will be directly executed. Note that a subroutine must return to the CALLing routine for SPRSTP to keep control. If this is not the case control may be transferred to the subject program. The next time (SHIFT) * is pressed the "*" will go out and CALLs and RSTs will be single-stepped, etc.
- (SHIFT) # Works like in control point 1; SPRSTP registers are loaded with TBUG registers. Control is returned to control point 2.
- @ @@s the SPRSTP registers AF'-IY. SP is retained.
- The following keys assist local writing and editing of 280 machine language strings and programs in RAM.
- ↑ Backspace. Advances the memory display to the predecessor location.
- ↓ Relative Locator. Displays address and contents of the location relative, in a twos complement sense, to the current location. If the current memory contents byte were the offset byte of a relative jump JR or DJNZ instruction the ↓ Relative Locator key shows where the transfer would go.
- (SHIFT) ↓ Relative Space. Advances memory display to the actual relative location. Also stores the current location for reference by...
- (SHIFT) ↑ Reference Space. Advances memory display to the location stored by (SHIFT) ↓ Relative Space. Used to return from working on a relative routine, or to return to a reference location established by the sequence (SHIFT) ↓, (SHIFT) ↑, such as the top of the code string you are writing, etc.

The following keys display and reformat RAM locations and contents.

- (SHIFT) ? Displays the ASCII or graphics character associated with the current contents byte.
- ;
- Line display of the 10H bytes starting with the current memory location. Displays the checksum of the row as =bb=. Advances memory to location 10H hence. The line can be displayed in either hex or ASCII, set by...
- (SHIFT) + Set hex/ASCII line. As loaded the line will be displayed in hex; after (SHIFT) + the line will be displayed in ASCII equivalents until the next time (SHIFT) + is pressed, when the line displays as hex, etc. etc.
- (SHIFT) < Delete current byte, pull successor string one location toward current location. Successor string is defined down to the user-placed FFFF string delimiter. You must place that FFFF delimiter. Don't use these keys until you place FFFF at the foot of the string you are working with. Illustrated:

```
# M 7000 00=09= 00 01 02 03 FF FF 06 07 FF FF FF FF FF FF FF FF
# M 7000 00      (SHIFT) Delete.
# M 7000 01=08= 01 02 03 FF FF FF 06 07 FF FF FF FF FF FF FF FF
```

The code string has moved toward the current location by one, leaving an FF into the foot of the string.

- (SHIFT) > Insert a byte, pushes string from current location to FFFF string delimiter one move away from the current location, rotates the byte wiped out by the leading FF into the current location for inspection. Illustrated:

```
# M 7000 01      (SHIFT) Insert.
# M 7000 FF=08= FF 01 02 03 FF FF 06 07 FF FF FF FF FF FF FF FF
# M 7000 FF      (SHIFT) Insert.
# M 7000 06=08= 06 FF 01 02 03 FF FF 07 FF FF FF FF FF FF FF FF
```


MISCELLANEOUS:

If you wish to experiment with the RST instructions, the high byte of the RST "page zero" is located within SPRSTP at location 5EC0H. As loaded this byte is initially 00. Insert your own high byte if you wish a programmable page zero. Of course the RSTs will only vector to your location under Single-step or TRACE, and when the (SHIFT) * CALL/RST status is set to single-step. As to why anyone would be interested in the RSTs, see my short article "Z80 User Stack Emulation" in BYTE magazine, January, 1980.

In the automatic mode the RAM Window will operate upon instructions such as indirect loads and stores, indirect I/O operations, indirect arithmetic and logical instructions, all indirect indexed instructions, successful JPs, JRs, CALLs, DJNZ and all block instructions. Here is a procedure using CPIR as a memory search function:

Set up the A register with some desired byte and HL with the address where you want the search to start. Use the (SHIFT) R register change key for this. Make sure RAM Window is on ← mode.

# M 7000 ED CPIR	<u>SPACEBAR</u>	
7001 B1	<u>SPACEBAR</u>	Note the RAM Window.
7002 FF	↑ Backspace	
7001 B1	↑ Backspace	
7000 ED CPIR	<u>SPACEBAR</u>	etc. etc.

Obviously the block comparisons and block loads can be used flexibly to compose the usual monitor MOVE and SEARCH functions.

The following illustrates the TRACE mode and the foreground 76-HALT breakpoint.

# 7000 CD CALL 45E9	TBUG routine puts keyboard byte in A.
7001 E9	
7002 45	
7003 FE CP 0D	Is it ENTER?
7004 0D	
7005 20 JR 7000	If not, check again.
7006 F9	
7007 76	If so, HALT the TRACE.

Set up this code and use TRACE starting at 7000. You can vary speed using 1 and 2 or HALT using Z or in this case, ENTER.

DANGEROUS BENDS:

Not removing the 76 HALT foreground breakpoint before # J jumping to the routine.

Using (SHIFT)< Delete and (SHIFT)> Insert without placing the FFFF string delimiter at the foot of the code string.

Loads of any kind into working RAM, especially block loads with large values in BC.

Subroutines that never return under (SHIFT) * execution.

Programs that test RAM by successively altering bytes, such as ROM and EDTASM. By the pigeon-hole principle working RAM in SPRSTP will be altered if you TRACE a program like this without care.

I/O instructions can enable flip-flops inopportunately.

Relocating TBUG without hitting the BREAK key. See below, STELEGS.

Don't let any of the above deter you from experimentation though, because often system crashes are educational. Also, the single key commands used on SPRSTP may take a little getting used to, since they are necessarily a little arbitrary, but it is the fastest and easiest command mode when familiar. So become a Super STEP adept!

STELEGS RELOCATOR: The top 300H bytes of SPRSTP are uncondensable relocater tables compatible with Super TLEGS (No. LL-0) Relocator for TBUG. Relocation procedures will be familiar to Super TLEGS users. In the following example we assume a linked TBUG-TLEGS-SPRSTP unit at 4380-4980, 49A0-4AA0, and 4B00-68FF.

First relocate TBUG. We will be using 32K resources here.

```
# BREAK          Delink TBUG and SPRSTP
# M 4831 80      Load DE with desired TBUG start address.
4832 83
4833 80          Load HL with present TBUG start address.
4834 43
# R              Check the TBUG registers.
# J 49A0         # J jump to Super TLEGS. Control is now
                  with the high TBUG.
```

We are ready to relocate SPRSTP. The entry point will be at 4CA0H. First we load DE and HL.

CONSTRAINT: DE must have 1E00H bytes vertical clearance.
E must be 00.

```
# M 8831 00      Load DE with desired SPRSTP start address.
8832 8B
8833 00          Load HL with present SPRSTP start address.
8834 4B
8835 80          IX loaded with desired linking TBUG start.
8836 83
8837 80          IY loaded with present linking TBUG start.
8838 43
# R              Check TBUG registers.
# J 4CA0         # J jump to STELEGS.
↓ S              Press S to link. ↓ is high copy prompt character.
```

To recap, STELEG needs the following parameters passed to it via the TBUG registers: DE contains desired SPRSTP starting address, with low byte 00, HL contains present SPRSTP starting address. IX contains the starting address of the copy of TBUG with which the new SPRSTP must link, and IY contains the starting address of the present copy of TBUG with which the present copy of SPRSTP is linked. Only when this material is entered can a successful # J 4CA0 be made.

ALLEN GELDER SOFTWARE
Box 11721 Main Post Office
San Francisco, CA 94101